

תאריך: 28.11.25

מבוא למדעי המחשב – כיתה יא-1

דף עבודה – מבוא לרשימות מקושרות (Linked Lists)

מטרות התרגיל

בפעילות זו תכירו לראשונה את מבנה הנתונים "רשימה מקושרת (Linked List)".
העבודה תתבצע צעד-אחר-צעד, וכל חלק מממש פעולה קטנה, כדי שתלמדו:

- מהי חוליה (Node)
 - איך יוצרים שרשרת חוליות
 - איך מצרפים חוליה בתחילת הרשימה
 - איך מצרפים חוליה בסוף הרשימה
 - איך עוברים על הרשימה וסופרים איברים
 - איך מחפשים ערך ברשימה
 - איך ניתן להדפיס את הרשימה בסדר הפוך על ידי שימוש ברקורסיה (משימת בונוס).
- במהלך העבודה תראו בצורה מוחשית איך משתנה מחלקה שנקרא "next" מחבר את החוליות זו לזו, ותוכלו לשחק עם הרשימה שלכם בהרבה דרכים.
- הדוגמא שנראה כאן היא **בסיסית, למטרת לימוד ראשונית ויכולת תכנות בסיסית ברשימות**.
לאחר מכן, בשיעורים ובתירגולים הבאים, נפתח את זה בהדרגה לעבודה "כללית" יותר עם רשימות, כפי שנדרש בבגרות. מאוד חשוב ללמוד ולהשקיע בכל שלב – זה יבנה אצלכם הבנה עמוקה.

הקבצים שאתם עובדים איתם

הקבצים הדרושים נמצאים ב GDRIVE. תוכלו להגיע אליהם בקלות דרך הקישור הבא:
[/https://tzvimelamed.com/lab](https://tzvimelamed.com/lab)

בתיקייה שקיבלתם יש את הקבצים:

- `Program.cs` – מכיל את ה-`Main` ואת התפריט
- `Node.cs` – מייצג חוליה בודדת
- `Utils.cs` – הדפסת צבעים וצלילים (ללא צורך לשנות)
- `ListTasks_Student.cs` – כאן נמצאות המשימות שאתם משלימים

שימו לב:

מוגדרות 7 פונקציות בשם `ListTaskX` (כלומר: `ListTask1()`, `ListTask2()`, ...).
כל פונקציה מייצגת שלב קטן ונפרד.

הנחיות כלליות

1. אל תשנו את מחלקת `Node` או `Utils`.
הפוקוס הוא על הבנת מבנה הרשימה ולא על עיצוב המחלקות.
2. כל פונקציה שאתם מממשים צריכה להשאיר את הקוד **פשוט, נקי, וקריא**.
3. בכל פעם שתסיימו משימה – הריצו את התוכנית ובדקו שהתוצאה נכונה.

החומר נועד לשימוש אישי של תלמידי תיכון קריית שרת בלבד.

משימה 1 – יצירת רשימה מוכנה מראש (A→B→C→D)

בפונקציה: `ListTask1()` - צרו 4 חוליות:

A, B, C, D

וקשרו אותן כך: `A → B → C → D → null`

החזירו את ראש הרשימה. (A)

משימה 2 – יצירת רשימה חדשה לגמרי (4 אותיות מהקלט)

בפונקציה `ListTask2()`:

בלולאה שהוכנה עבורכם כבר מתקבלות 4 אותיות מהמשתמש.

השלימו:

- יצירת Node חדש עבור כל אות
 - הכנסת החוליה בתחילת הרשימה (insert at head)
- נסו להבין מה יהיה הסדר הסופי...

משימה 3 – הוספת חוליה בתחילת הרשימה (HEAD INSERT)

בפונקציה `ListTask3(Node head)`

קבלו קלט - אות מהמשתמש,

יצרו חוליה חדשה,

והכניסו אותה בתחילת הרשימה:

`newNode → oldHead → ...`

החזירו את הראש החדש.

משימה 4 – הוספת חוליה בסוף הרשימה (TAIL INSERT)

בפונקציה `ListTask4(Node head)`

קבלו אות מהמשתמש,

צרו חוליה חדשה,

עברו על הרשימה עד לסוף,

וחברו אותה שם.

חשוב: אל תשנו את הראש!

משימה 5 – ספירת מספר החוליות

בפונקציה `ListTask5(Node head)`

עברו על הרשימה בעזרת משתנה עזר (pointer) וספרו כמה חוליות יש.

הדפיסו את התוצאה.

משימה 6 – חיפוש ערך ברשימה

בפונקציה `ListTask6(Node head)`:

- קבלו אות מהמשתמש.
- עברו על הרשימה וחפשו אותה.
- אם נמצא – הדפיסו את מיקום החוליה.
- אם לא – הדפיסו הודעה מתאימה.
- טיפ: המיקום הראשון הוא 1 (לא 0).

משימה 7 Bonus – הדפסה הפוכה (רקורסיה)

בפונקציה `ListTask7(Node head)`:

- מטרת המשימה:
- להדגים שימוש רקורסיבי בנתון רקורסיבי – רשימה מקושרת.
- השלימו את הרעיון:
- אם הרשימה ריקה – עצירה
- אחרת – הדפס הפוך את הרשימה מהקצה (קריאה רקורסיבית)
- ואז הדפס את החוליה הנוכחית

שאלות הבנה ועיון (יתווספו כאן עוד שאלות)

1. מה החשיבות משתנה `next` ?
2. איך ליצור רשימה מאפס ?
3. מה ההבדל בין "ראש הרשימה" לבין שאר החוליות ?
4. איך עוברים בלולאה על רשימה ?
5. איך מחפשים ערך ? (כלומר האם ערך מסוים נמצא ברשימה)?
6. למה רקורסיה מתאימה לפעמים לפעולה על רשימה מקושרת? באיזה מקרים נצטרך רקורסיה ובאיזה מקרים נצטרך לולאה?
7. מה לדעתכם היתרון של רשימה מקושרת על פני מערך? למה צריך את זה בכלל?
8. האם נהניתם מהתרגיל ?

שיעורי בית:

- א. השלימו את המשימה בבית.
- ב. המציאו לעצמכם עוד שתי משימות ועדכנו את התכנית (לדוגמא:
 - a. כתבו משימה שעוברת על הרשימה וכותבת את האיברים בדילוגים, כלומר כותבת את הראשון ומדלגת על השני, כותבת את השלישי ומדלגת על הרביעי).
 - b. המציאו אחת נוספת.